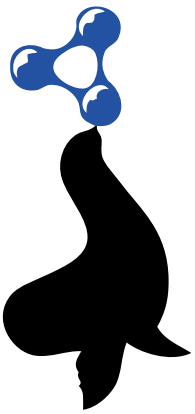
Evaluation design and collection of test data for matching tools (v2)

Christian Meilicke, Cassia Trojahn dos Santos, Heiner Stuckenschmidt, Maria Rosoiu



SEALS

Semantic Evaluation at Large Scale

FP7 – 238975

D12.4 Evaluation Design and Collection of Test Data for Matching Tools - v2

Coordinator: Christian Meilicke
With contributions from: Cássia Trojahn dos Santos, Heiner Stuckenschmidt, Maria Rosoiu
Quality Controller: Jérôme Euzenat
Quality Assurance Coordinator: Raúl García Castro

Document Identifier:	SEALS/2010/D12.4/V2.0
Class Deliverable:	SEALS EU-IST-2009-238975
Version:	version 2.0
Date:	May 20, 2011
State:	final
Distribution:	public

Executive Summary

The lessons learnt from the experiences gathered in the first evaluation campaign [19] as well as the development progress of the SEALS platform allows us to improve and extend the design of the second evaluation campaign. This campaign is planned to take place in the context of OAEI, next in the summer of 2011. This deliverable reports the design of this campaign, highlighting the main extensions and novelties:

- **new test data** (Chapter 4): in the first campaign, only three OAEI test cases (anatomy, benchmark and conference) have been used and we plan to add more datasets from different resources, namely directory, multilingual conference and UMLS meta-thesaurus.
- **automatic test generation** (Chapter 4): the benchmark test case is not discriminant enough between systems and we plan to introduce a controlled automatic test generator for creating tests of increasing difficulty.
- **new evaluation criterion** (Chapter 3): in the first campaign, due to the lack of a controlled execution environment, the evaluation criterion was mostly conformance to reference alignments. Now that core parts of the SEALS platform have been delivered, we plan to evaluate efficiency, mainly in terms of runtime.

We also present the evaluation workflows that represent the interaction between evaluation components in an evaluation scenario (Chapter 2). We have identified three different evaluation workflows. Finally, we have selected a set of potential targets, based on maturity and availability (Chapter 5): Aflood, AgreementMaker, Aroma, Eff2Match, Falcon, Lily, Rimon, Sobom and Taxomap. We briefly report on our tests of running these tools on the express runtime version of the platform.

Document Information

IST Project Number	FP7 – 238975	**Acronym**	SEALS
Full Title	Semantic Evaluation at Large Scale		
Project URL	`http://www.seals-project.eu/`		
Document URL			
EU Project Officer	Carmela Asero		

Deliverable	**Number**	12.4	**Title**	Evaluation Design and Collection of Test Data for Matching Tools - v2
Work Package	**Number**	12	**Title**	Matching Tools

Date of Delivery	**Contractual**	M24	**Actual**	31-05-11
Status	version 2.0		final ☒	
Nature	prototype ☐ report ☒ dissemination ☐			
Dissemination level	public ☒ consortium ☐			

Authors (Partner)	Christian Meilicke (University Mannheim), Cássia Trojahn dos Santos (INRIA), Heiner Stuckenschmidt (University Mannheim),			
Resp. Author	**Name**	Christian Meilicke	**E-mail**	christian@informatik.uni-mannheim.de
	Partner	U. Mannheim	**Phone**	+49 621 181 2484

Abstract (for dissemination)	Based on the results of the first evaluation campaign (T12.3), and taking into account the technical progress of the SEALS platform, we deliver an updated and extended evaluation and test data design for our second evaluation campaign. This campaign is planned to take place in the context of the OAEI at the ISWC 2011.
Keywords	ontology matching, ontology alignment, evaluation, benchmarks, efficiency measure

Version Log			
Issue Date	**Rev No.**	**Author**	**Change**
01/02/2011	1	Christian Meilicke	Set up document with content from D12.1
03/04/2011	2	Christian Meilicke	Revised document structure
03/04/2011	3	Christian Meilicke	Wrote sections on tools and criteria
04/04/2011	4	Cassia Trojahn	Wrote executive summary and introduction
04/04/2011	5	Cassia Trojahn	Wrote evaluation workflows section
04/04/2011	6	Cassia Trojahn	Added bpel annexe
05/04/2011	7	Christian Meilicke	Added summary and content to data section
06/04/2011	8	Cassia Trojahn	Revised/extended whole doument
25/04/2011	9	Christian Meilicke	Included comments of reviewer

Project Consortium Information

Participant's name	Partner	Contact
Universidad Politécnica de Madrid		Asunción Gómez-Pérez Email: asun@fi.upm.es
University of Sheffield	The University Of Sheffield.	Fabio Ciravegna Email: fabio@dcs.shef.ac.uk
Forschungszentrum Informatik	FZI	Rudi Studer Email: studer@aifb.uni-karlsruhe.de
University of Innsbruck	STI · INNSBRUCK	Barry Norton Email: barry.norton@sti2.at
Institut National de Recherche en Informatique et en Automatique	INRIA	Jérôme Euzenat Email: Jerome.Euzenat@inrialpes.fr
University of Mannheim	UNIVERSITÄT MANNHEIM	Heiner Stuckenschmidt Email: heiner@informatik.uni-mannheim.de
University of Zurich	University of Zurich Department of Informatics DIS Dynamic and Distributed Information Systems	Abraham Bernstein Email: bernstein@ifi.uzh.ch
Open University	The Open University	John Domingue Email: j.b.domingue@open.ac.uk
Semantic Technology Institute International	STI · INTERNATIONAL	Alexander Wahler Email: alexander.wahler@sti2.org
University of Oxford	UNIVERSITY OF OXFORD	Ian Horrocks Email: ian.horrocks@comlab.oxford.ac.uk

Table of Contents

List of Figures

List of Tables

1. Introduction

OAEI and SEALS have been coordinated since 2010 in order to design principled and reproducible evaluation techniques for matching tasks. The first SEALS/OAEI evaluation campaign of ontology matching systems has been carried out in autumn 2010. The design of this coordinated campaign, in terms of methodology, evaluation criteria, datasets and targeted tools, was reported in [6]. The crucial impact was the introduction of a technology for automating the evaluation process, which has affected both tool developers and campaign organizers [19].

The progressing maturity of the SEALS platform enables us now to use an extended approach. In particular, we will switch from a web based evaluation scenario [18] to a scenario where we execute the systems on top of the platform. This, as well as the lessons learned from OAEI 2010, has an influence on several aspects related to evaluation design, criteria to be measured, test data, and required preparation.

The deliverable is structured as follows. In Chapter 2, we present an extended version of the evaluation workflow we have designed for the first campaign, mainly in terms of new evaluation criteria and inputs for the matching process. Chapter 3 presents the evaluation criteria to be taken into account for evaluating systems. In Chapter 4, we present the evaluation test cases that we will use, highlighting potential new test cases and the automatic generator of tests. A list of potential evaluation targets and a short report on testing them is presented in Chapter 5. Finally, we conclude the deliverable in Chapter 6.

2. Evaluation Workflows

The goal of this chapter is to present the evaluation workflows that represent the sequence of activities carried out in different evaluation experiments. In [6], we have presented a basic evaluation workflow and its extensions, in order to consider the lack of reference alignments or the intervention of the user in the evaluation loop. In this deliverable, we focus on the specific workflows that will be considered in the second campaign. First, in Section 2.1, we present the basic workflow, which has been already used in the first campaign and next (Section 2.2) we present its extensions.

2.1 Basic workflow

An evaluation workflow represents the interaction between several components in an evaluation experiment (matchers, test providers, evaluators, etc.). The basic workflow restricts the experiment to the evaluation of one matcher using a set of test cases (Figure 2.1), using conformance as evaluation criterion.

As illustrated in Figure 2.1, the first step is to retrieve, from a repository of *tests*, the ontologies to be matched and the corresponding reference alignments. These are the components of the test cases to be considered in such an evaluation. Next, the *matcher* performs the *matching* task, taking the two ontologies as input parameters. Then, the resulting alignment is evaluated against the reference alignment, by an *evaluator*. Finally, each raw result (alignment A) and its interpretations i, i.e., precision and recall, are stored into the *result* repository.

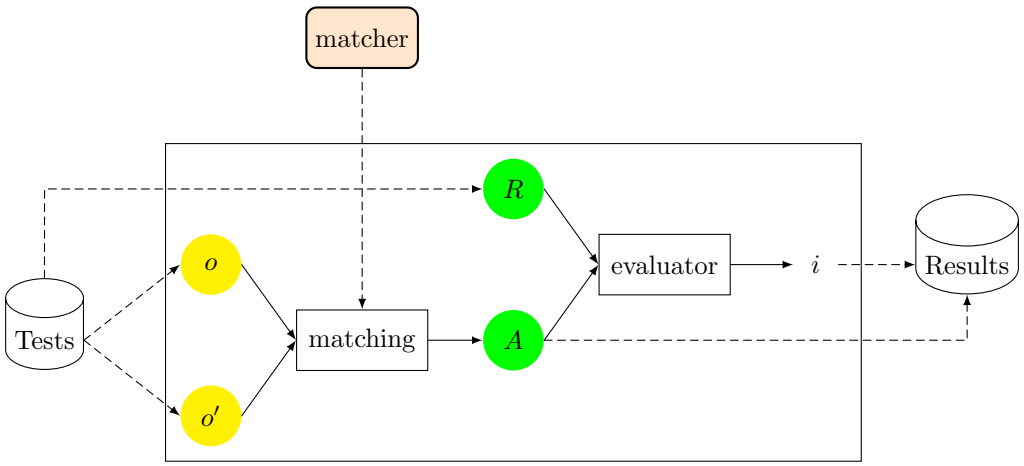


Figure 2.1: Basic evaluation workflow.

In Appendix A, an executable BPEL workflow, that corresponds to the workflow of Figure 2.1, can be found.

2.2 Extended workflows

Due to the variability of the alignment evaluation, different scenarios can be specified, by adding new components to the process presented in Figure 2.1. We have identified two main extensions, by taking into account:

Use of an input alignment. In some test cases (e.g., subtask #4 of the OAEI anatomy track), an initial alignment is required by the matching process. Thus, three input parameters must be taken: the two ontologies and an initial alignment.

Test generator. Test cases can be generated from a description of the kind of evaluation to be executed (for example, removing n% of the properties of the ontologies). A description of the desired test case must be provided and the output of the test generation process is then used as input to the matching process.

Figure 2.2 illustrates the first case, where an input alignment A' is taken into account. This alignment comes from the test repository together with the reference alignment and input ontologies.

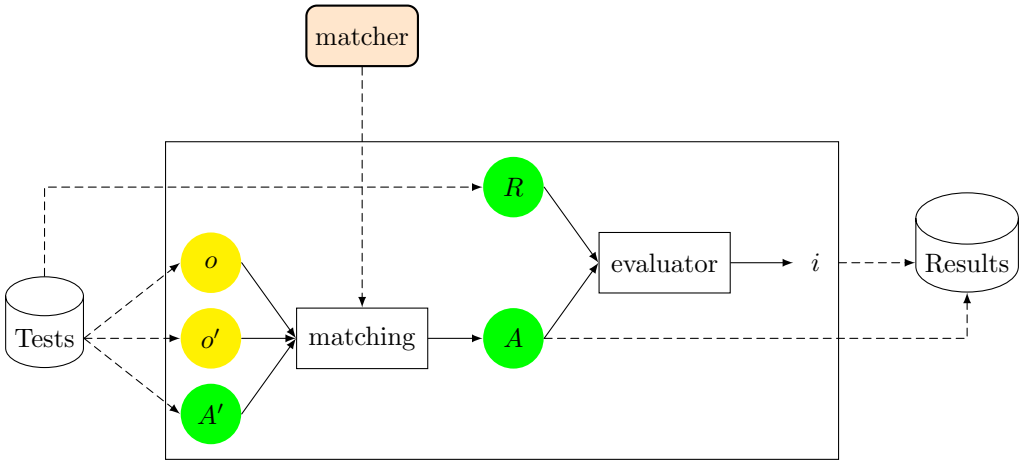


Figure 2.2: Extended evaluation workflow with alternative input alignment.

Regarding the generation of tests, Figure 2.3 illustrates a more elaborated workflow where the test cases are automatically generated by a test generator, according to some description provided by the user and based on some reference ontology. As output, the generation process creates a set of alternative ontologies, from a reference ontology, for instance, by removing its properties or individuals or adding new subclasses into the ontology hierarchy.

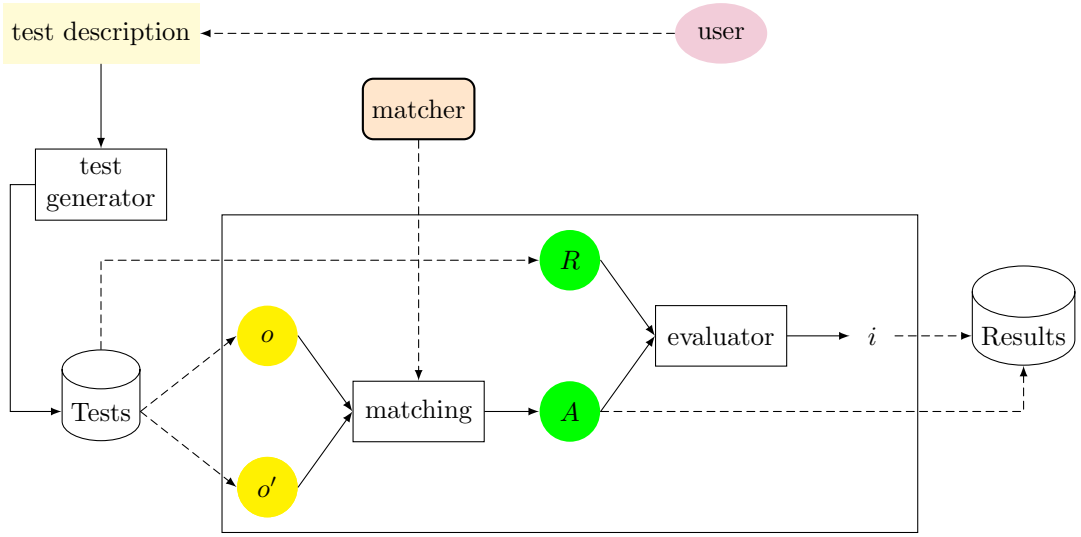


Figure 2.3: Extended evaluation workflow with test generation.

3. Criteria and Measures

In the following, we give a concise overview on the criteria that we plan to use in the context of the second evaluation campaign. Note that we gave a comprehensive overview on different criteria and metrics already in [6]. We used most of these metrics, except effiency and scalability related metrics, already for the OAEI 2010.

3.1 Interoperability

Despite efforts on composing matchers [13] and on defining an Alignment API [7], ontology matching lacks interoperability benchmarks between tools. The first attempt to evaluate interoperability[1] between ontology matching systems is to measure their compliance to standards such as RDF(S) and OWL. In accordance to this, we will test whether systems are able to correctly work on ontologies specified in the language standard RDF(S) and OWL.

We do not use a specific way to measure non-conformance to these standards, nor do we use a specific dataset for this purpose. Non-conformance can be detected indirectly through different criteria. For instance, the inability to identify class names in a certain language will lead to a decrease of recall. The inability to parse a certain ontology might results in throwing an error and the matching tool will eventually terminate without returning any results. We have to take care of these issues, both in the definition of the workflow (appropriate error-handling) and when we finally generate a report from the evaluation results. In particular, we will count the number of testcases that resulted in an internal error thrown by the matching system as well as the number of testcases that resulted in an empty alignment. In the latter case we have to distinguish between those cases where the matching systems failed to read the relevant information for syntactic reason and the cases where the matching system generated an empty alignment on purpose.

3.2 Compliance against a reference alignment

There are many ways to qualitatively evaluate returned results [5]. One possibility is to propose a reference alignment R that is the one that the participants must find (sometimes also called a *gold standard*). The alignment A generated by the evaluated alignment algorithm can then be compared to that reference alignment.

The most commonly used measures are precision (true positive/retrieved) and recall (true positive/expected) which have been adopted for ontology alignment. They are commonplace measures in information retrieval.

Definition 1 (Precision) *Given an alignment A and a reference alignment R, the precision of A is given by*

$$P(A,R) = \frac{|R \cap A|}{|A|}.$$

[1] `http://knowledgeweb.semanticweb.org/benchmarking_interoperability/owl/index.html`

the recall of A is given by

$$R(A, R) = \frac{|R \cap A|}{|R|}.$$

In addition to precision and recall, we will also use the f-measure, which is defined as the weighted harmonic mean of precision and recall. The libraries that compute precision, recall, and f-measure are already available as part of the Alignment API [7].

Other measures are available withing the Alignment API, such as relaxed and semantic versions of precision and recall as well as confidence weighted precision and recall which have been used in OAEI 2010.

3.3 Efficiency and scalability

Metrics such as execution time (speed) and amount of required memory are usually considered to measure efficiency. For the first campaign we planned to measure these aspects of a matching process, however, the platform was not yet available. We had to use our web-based evaluation service to conduct an automated evaluation campaign. In this setting, the measurement of runtimes or required memory was not possible, since all tools have been executed on different machines.

Still none of these measurements (related to both runtime and memory) are implemented in the platform. Note that these kinds of measurements cannot be implemented as work-package specific services. They belong to a generic type of functionality that is required by each workpackage in the same way.

Without evaluating the efficiency of matching systems we loose a unique selling point, especially in comparison to the web-based evaluation approach of 2010. We had many requests by tool vendors related to this functionality. In particular, the measurement of runtimes was requested. Thus, we have to find a way to implement these metrics in time for using the SEALS platform in the context of the OAEI 2011 (or 2012 at the latest).

Scalability is a metric that deals with the dependency between size of the input problem (e.g., size of the ontologies to be aligned) and the resources needed to perform the matching operation. It is clear that we can only measure scalability of a tool, if we can measure its efficiency for a specific test case. Implementing the missing functionality is thus also crucial for scalability.

4. Test Data for Evaluation

This chapter describes the potential datasets we plan to use in the second campaign. In the first three sections we briefly describe the datasets we already used in 2010. In the remaining sections we describe potential new datasets. For these datasets, we have not yet decided whether we can already use them for the OAEI 2011. In addition, we describe a new test data generator.

4.1 Benchmark

The systematic benchmark test set is built around one reference ontology and many variations of it. The ontologies are described in OWL-DL and serialized in the RDF/XML format. The reference ontology is that of test #101. It contains 33 named classes, 24 object properties, 40 data properties, 56 named individuals and 20 anonymous individuals. Participants have to match this reference ontology with the variations. Variations are focused on the characterization of the behavior of the tools rather than having them compete on real-life problems. They are organized in three groups:

Simple tests (1xx) such as comparing the reference ontology with itself, with another irrelevant ontology (the wine ontology used in the OWL primer) or the same ontology in its restriction to OWL-Lite;

Systematic tests (2xx) obtained by discarding features from some reference ontology. It aims at evaluating how an algorithm behaves when a particular type of information is lacking. The considered features were:

Four real-life ontologies of bibliographic references (3xx) found on the web and left mostly untouched (there were added xmlns and xml:base attributes).

The dataset has been used since several years in the OAEI with only minor modifications. We gave a more detailed description in Deliverable D12.1 [6]. We supported this track in 2010. In 2011, we plan to run this track on the SEALS platform.

4.2 Anatomy

The ontologies of the anatomy track are the NCI Thesaurus describing the human anatomy, published by the National Cancer Institute (NCI)[1], and the Adult Mouse Anatomical Dictionary[2], which has been developed as part of the Mouse Gene Expression Database project. Both resources are part of the Open Biomedical Ontologies (OBO). They are typical examples of large, carefully designed ontologies that are described in technical terms.

We gave a more detailed description of the dataset in Deliverable D12.1 [6]. The anatomy track is the only track that makes use of an additional input alignment. While in 2010 we abstained from including this aspect within the SEALS approach, we plan in 2011 to include this in the automated evaluation supported by SEALS.

[1] `http://www.cancer.gov/cancerinfo/terminologyresources/`
[2] `http://www.informatics.jax.org/searches/AMA_form.shtml`

Ontology	Type	Concepts	Datatype Prop.	Object Prop.	Expressivity	Ref
Ekaw	Insider	77	-	33	$\mathcal{SHIN}$	Yes
Sofsem	Insider	60	18	46	$\mathcal{ALCHIF}$(D)	Yes
Sigkdd	Web	49	11	17	$\mathcal{ALEI}$(D)	Yes
Iasted	Web	140	3	38	$\mathcal{ALCIN}$(D)	Yes
Micro	Web	32	9	17	$\mathcal{ALCOIN}$(D)	-
Confious	Tool	57	5	52	$\mathcal{SHIN}$(D)	-
Pcs	Tool	23	14	24	$\mathcal{ALCIF}$(D)	-
OpenConf	Tool	62	21	24	$\mathcal{ALCOI}$(D)	-
ConfTool	Tool	38	23	13	$\mathcal{SIN}$(D)	Yes
Crs	Tool	14	2	15	$\mathcal{ALCIF}$(D)	-
Cmt	Tool	36	10	49	$\mathcal{ALCIN}$(D)	Yes
Cocus	Tool	55	-	35	$\mathcal{ALCIF}$	-
Paperdyne	Tool	47	21	61	$\mathcal{ALCHIN}$(D)	-
Edas	Tool	104	20	30	$\mathcal{ALCOIN}$(D)	Yes
MyReview	Tool	39	17	49	$\mathcal{ALCOIN}$(D)	-

Table 4.1: Ontologies of the conference test set.

4.3 Conference

The conference dataset consists of a collection of ontologies that describe the same domain, namely the domain of conference organization. This dataset has been developed by a group of researchers from the University of Economics, Prague. Its origin is described in [17]. The characteristics of the dataset, which consists of 15 ontologies, are listed in Table 4.1. In 2010, we already supported the evaluations related to the conference dataset. We plan to do this again in 2011.

4.4 Directory

So far we supported the evaluations related to three important OAEI datasets in 2010 and we will do this again for the OAEI 2011. The next OAEI dataset that we plan to support is the dataset of the directory track. The focus of this track is to evaluate precision and recall in a real world taxonomy integration scenario. Within this track the organizers analyze whether ontology matching tools can effectively be applied to the integration of *shallow ontologies* as we find them for web directories. The concrete datasets were extracted from Google, Yahoo and Looksmart web directories. The specific characteristics of the datasets are simple relationships as well as vague terminology and modeling principles.

The dataset, thus, seems to be a perfect complement compared to the benchmark, conference, and anatomy datasets, which consist of (mainly) well-modeled ontologies with higher expressivity. We are currently in contact with the organizers of the directory track and discuss in how far the evaluations centered around this dataset can be supported by SEALS.

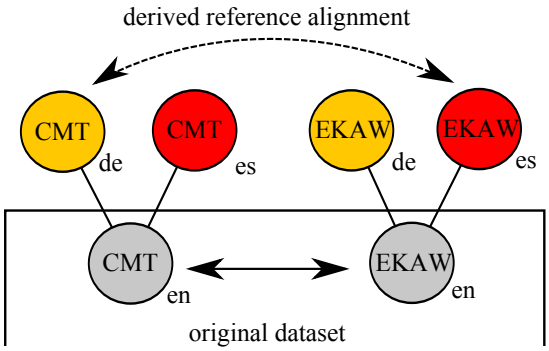


Figure 4.1: Generating a multilingual evaluation dataset

4.5 Multilingual Conference

All of our datasets are annotated with English names and labels (or technical terms from a specific domain)[3]. This neglects the fact that there are many ontologies, which are not described by / built from an English terminology. In the following we report on our attempts to create a comprehensive dataset, which deals with the important issue of multilingual ontology matching.

Our approach is based on an existing monolingual dataset, which comprises both a set of different ontologies as well as reference alignments between them. We have chosen the conference dataset, namely the subset of ontologies that comes with a reference alignment. Then we translated the ontologies in different languages. In particular, we planned to generate for each ontology a Spanish, German, French, Russian, Portuguese, Czech, Dutch, and Chinese version.

As illustrated in Figure 4.1, which depicts a small subset of the dataset, this dataset allows to derive a comprehensive set of non-trivial multilingual matching tasks. In particular, we can generate for each pair of languages (9 languages, 36 pairs of languages) a reference alignment between all ordered pairs of ontologies (7 ontologies, $2 \times 21 = 42$ unordered pairs). Overall we have thus 1512 non-trivial multilingual matching tasks based on the high quality reference alignments of the original dataset.

The current version of the dataset is available in raw-format at `http://webrum.uni-mannheim.de/math/lski/matching/trans/`. We have been organizing the development of this dataset together with a group of different researchers[4]. Not all translations have been finished yet. We analyse currently in how far we can use a fraction of the dataset for the 2011 campaign. A small sample subset of the dataset is already available as a test suite in the SEALS test repository at `http://seals.sti2.at/tdrs-web/testdata/persistent/ConferenceML+Testsuite`.

[3] An exceptions are two test cases from the benchmark dataset.

[4] The following persons have been involved in the development of the dataset: Elena Montiel-Ponsoda, Raúl García Castro, Dominique Ritze, Rim Helaoui, Andrei Tamilin, Fred Freitas, Ryan Ribeiro de Azevedo, Icaro Medeiros, Fernando Lins, Eric Rommel, Roberta Fernandes, Ondrej Zamazal, Vojtech Svatek, Willem Robert van Hage, and Shenghui Wang.

4.6 Linked Open Data Schemas

Currently a group of researchers from the Kno.e.sis Center at the Wright State University (USA) is working on a dataset for matching Linked Open Data (LOD) schemas [11]. This dataset will probably be used in a new OAEI track in 2011. We have contacted the developer of the dataset and offered to support them in conducting the evaluations for this new track on top of the SEALS platform.

The objective of this dataset is to demonstrate the capabilities of matching systems with respect to effectiveness in identifying subclass and equivalence relationship between classes of LOD schemas. Reference alignments between different LOD dataset schemas have been created manually. Furthermore, the dataset contains also alignments between several LOD datasets and the upper level ontology PROTON.

4.7 Further datasets

Aside from these datasets we have identified two additional ones. However, further analysis is required to finally decide whether these datasets are well-suited. A comprehensive dataset was proposed in [12]. The authors show that the UMLS Metathesaurus [1] (UMLS) can be used as a silver standard to align ontologies such as SNOMED CT, FMA or NCI. However, the authors also argue that UMLS contains a significant number of logic errors that have to be taken into account.

Another source of interesting datasets can be found at `http://www.obofoundry.org/`. This web page is hosted by the OBO foundry, which is an initiative involving developers of science-based ontologies who are establishing a set of principles for ontology development with the goal of creating a suite of orthogonal interoperable reference ontologies in the biomedical domain [16]. Based on these efforts, a comprehensive set of correspondences and so called bridges between different important biomedical ontologies are available for download. It has to be clarified in how far these correspondences have been verified by domain experts with respect to correctness and completeness.

4.8 Test data generator

In order to generate a systematic benchmark of increasing difficulty in an automatic and parameterized way, a test generator has been designed. It allows for simulating a variety of situations and evaluating how matchers face them.

Basically, the generator receives as input an ontology and a set of parameters related to the kinds of modifications that must be applied over that ontology (see Figure 4.2). It produces as output a set of modified ontologies together with the corresponding reference alignments (i.e., alignments between the initial and modified ontologies).

The following set of input parameters has been defined, which corresponds to the modifications that can be applied over the initial ontology:

- remove classes;

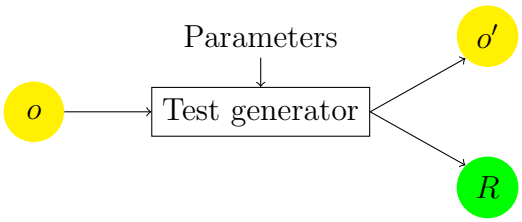


Figure 4.2: Generation process.

- remove properties;
- remove comments;
- remove restrictions;
- add classes;
- add properties;
- rename classes;
- rename properties.

The use of such a systematic test generator has several benefits over previous benchmarks. First, one can generate tests from different initial ontologies. Second, the tests can be generated following variable levels of granularity.

From the general scenario, we can have different variations:

- **individual test generation** (displayed in Figure 4.2): from an initial ontology and a set of input parameters, a test case, i.e., the modified ontology and reference alignment, is generated;
- **data set generation** (based on the former): from an ontology, a full data set (ontologies + reference alignment pairs) is generated. This data set can feature all variations of the parameters following a given stepwise increment or a predefined data set profile, e.g., the test profiles of the previous OAEI benchmarks.

Furthermore, the test generation can happen:

- **offline**: the previous benchmark data set is replaced by a newly generated dataset (this is what will be used for the next campaign);
- **online**: tests are generated on the fly (this requires a fine calibration of the test difficulty).

5. Evaluation Targets

Planning the first campaign (Deliverable D12.1 [6]), we focused on five systems as potential evaluation targets. As reported in D12.3 [19], finally 13 of 15 participants from the OAEI 2010 participated in one of the SEALS tracks. Among them three systems from the set of five systems that we originally described as potential evaluation targets. We can draw the conclusion that the first campaign was a success in terms of participation. Now we have to ensure that the same will be the case for the second campaign.

In the following, we introduce the modifications of the infrastructure that will – from the tool developers point of view – have an impact on participating in the second campaign (Section 5.1). We report on our efforts to ensure that these changes might not result in a barrier for participation and list the systems for which we already tested that technical hurdles will not be a problem (Section 5.2).

5.1 Background and infrastructure

For the first campaign many components have not yet been available in an integrated platform. For that reason we had to find a way to offer the services required to perform the automated evaluation in a different way. A detailed description of our approach can be found in Deliverable D12.2 [14] and in [18].

The approach we used for the first campaign was based on a web service interface wrapping the functionality of a tool to be evaluated. This interface allows to execute and evaluate the tool without the need for a runtime environment. Moreover, the tool has been executed on the machine of the tool developer, while the evaluation took place within the SEALS infrastructure. In this approach each tool developer had to implement a web service to make the functionality of his tool available to the SEALS infrastructure.

For the second campaign, the SEALS platform has evolved and can probably be used for *running the tool* on the platform itself. This requires the tool developer to implement a different interface, i.e. to follow a completely different approach. In particular, the tool has to be packaged together with all required dependencies. The SEALS platform can then execute the packaged tool taking into account all dependencies that have been made explicit.

All of the development and debugging required for running a tool via the SEALS online service was done by the tool developer. The same is currently not the case with respect to the interface for running the tool on the SEALS platform. In particular, one needs to deploy the SEALS platform or an express version of it to perform the final test for running the tool. This is in principle possible for a tool developer, however, the hurdle for doing so is at the moment still too high.

For that reason we are currently developing a tutorial and some tools/templates to support the interface implementation and tool packaging on side of the tool developer (sometimes briefly referred to as tool wrapping). More details on this will be available in the final version of Deliverable D12.5. While developing these documents and libraries, we tested them with a comprehensive set of tools. This process has been

Matching System	Size	2nd Campaign	Comment
AFlood [15]	6.1 MB	unknown	-
AgreementMaker [3]	18.4 MB	yes	tested with simple OAEI 2009 setting only
Aroma [4]	8.1 MB	unknown	-
Eff2match [2]	48.4 MB	yes	-
Falcon [10]	12.5 MB	unknown	-
Lily [20]	16.9 MB	yes	runs only on WIN, change language options to English(USA)
Rimom [21]	11.2 MB	yes	-
Sobom [22]	12.6 MB	unknown	-
Taxomap [9]	57.3 MB	yes	requires additional perl installation on windows

Table 5.1: Matching tools currently executable on top of the SEALS platform.

conducted partially in consultation with the developer of the specific tool. We report on our experiences briefly in the following section.

5.2 Wrapped Matching Systems

The results of our attempts to wrap some of the most important ontology matching tools are listed in Table 5.1. All of the systems listed there have been wrapped successfully. Most of them run on both Linux and Windows systems (only Lily runs on Windows only). Most systems require in addition to the included libraries some additional resources. For all of them it was sufficient to make the relevant files available from the working directory where the matching process is executed by the SEALS platform.

We have also asked some of the tool vendors whether they plan to participate in the OAEI 2011, and thus also in the second evaluation campaign. So far we have a positive answer from 5 systems. Some of the other systems we will eventually run on our own. Especially AFood and Aroma are known to be very efficient with respect to their runtime [8]. Their runtime results will thus be an important reference value.

With this set of systems we have reached a critical mass. Moreover, we used our experiences to improve documentation/material that supports tool vendors to wrap the tools on their own. During the next months we have to show that all components can be combined successfully by running a final integration test. In this test we will simulate an evaluation campaign on top of the platform using the set of matching systems in Table 5.1. Moreover, we have to finish our work on the material that supports the process of wrapping a tool. More details have to be made available in the final version of D12.5.

6. Conclusions

In this deliverable we described the current status of planning the SEALS support for OAEI 2011. The OAEI 2011 will be the second ontology matching evaluation campaign that is supported by SEALS technology. Our second campaign will thus take place earlier than it was planned originally. This decision is based on the importance and acceptance of the OAEI by the ontology matching community. We presented the approach taken for OAEI 2010 in a concise format and focused then on the modifications, extensions and differences relevant for the OAEI 2011, which are summarized in the following.

Criteria and Metrics. For the second evaluation campaign a limited set of criteria will be used that can be tested using the workflows as described in this deliverable. Criteria and measures to be considered are:

- Efficiency: runtime;
- Interoperability: compliance to the standard language RDFS and OWL-DL;
- Compliance with reference alignment: precision and recall;

Datasets. The datasets were selected based on the existence of reliable reference alignments and experiences with using the datasets in evaluation campaigns. Furthermore, we list a potential set of new datasets that exploit complementary features from those in the previous datasets.

- Anatomy (OAEI dataset already supported in 2010),
- Benchmark (OAEI dataset already supported in 2010),
- Conference (OAEI dataset already supported in 2010),
- Directory (OAEI dataset not yet supported)
- LOD schemas (probably new OAEI track),
- Multilingual conference (newly generated dataset),
- Diverse biomedical test cases (available datasets to be analyzed),
- Generated test cases (new generator to create datasets).

However, further analysis is required to ensure, for example, the correctness and completeness of the reference alignments for some of these testsets.

Tools. For the second campaign, we have selected a subset of the systems that have been involved in past OAEI campaigns. Tools have been selected based on maturity and availability. Based on these criteria, we have identified the following tools as potential candidates to participate in the second campaign:

- AFlood,
- AgreementMaker,
- Aroma,
- Eff2Match,
- Falcon,

- Lily
- Rimom
- Sobom,
- Taxomap.

We have as well explained the technical differences between the SEALS support for OAEI 2010 and our plans for 2011. We focused on the consequences for participants. In particular, we reported on our successful tests of running nine matching tools on the SEALS platform[1]. The lessons learned from this effort will finally be summarized in a tutorial to support tool developers to deploy and execute their tools on the SEALS platform autonomous.

[1]We refer here to the SEALS express runtime version 1.0

A. BPEL workflow

Figure A.1 illustrates the workflow for evaluating a matching tool. It iterates over a test suite and invokes a tool that generates an alignment between two ontologies. After receiving the input parameters (`receiveInput`), the result composer metadata is specified (`assignMetadata` and `addMetadaResults`), the parameters for loading a test suite are initialised (`setParamsToLoadTestSuite`) and the test suite is loaded (`loadTestSuite`).

In ontology matching, one test suite is composed of a set of ontology pairs and their reference alignments. Each test case in the test suite contains three data items: ontology source, ontology target and reference alignment. The `getOntoSource` and `getOntoTarget` activities retrieve the URLs of the corresponding data items from the Test Data repository.

In order to invoke the tool (`align`), the URLs of the two ontologies are used as input parameters. The result of this invocation is the URL of the file containing the generated alignment. This URL is then sent to the results composer (`addRawResultItem`), which will later use this information to generate the ZIP file containing all generated raw results. The output of the tool together with the reference alignment (`getRefAlign`) are sent to the interpreter (`interpret`), which computes precision and recall. Next, the interpretations are added to the results composer bundle (`addInterpretationsItem`). This process is repeated for each test case. After that, ZIP files containing the raw results (`createBundleRawResults`) and interpretations (`createBundleIntepretations`) are generated by the results composer service and these bundles are submitted to the results repository (`addRawResults` and `addInterpretations`, respectively).

Particular fragments of the BPEL workflow (Figure A.1) are dedicated to fault handling. Service interfaces specify the faults that can be returned by each operation. A generic business fault, `SEALSFault`, has been specified for wrapping the faults that occur when executing these operations. This generic fault wraps the real faults: tool faults and unsupported platform faults. On the other hand, when implementing the tool interface, for instance, developers have to wrap all the tool exceptions into the `SEALSFault`.

Three ways to handle faults have been considered within the workflow. For instance, at the invoke level (`align` activity), it is verified if the execution of the tool raises one of the tool faults, i.e., `ToolBridgeFault`, related to the tool interface or `ToolFault`, related with the execution of the tool itself. At the scope level, the execution of activities subsequent to a tool execution fault, can be skipped. Finally, at the process level, a global fault handler is responsible for catching the `SEALSFault` and verifying if the nested fault is an unsupported platform fault (`UnexpectedPlarformFault`). In this case, the process instance ends immediately calling the callback service with an unsuccessful response.

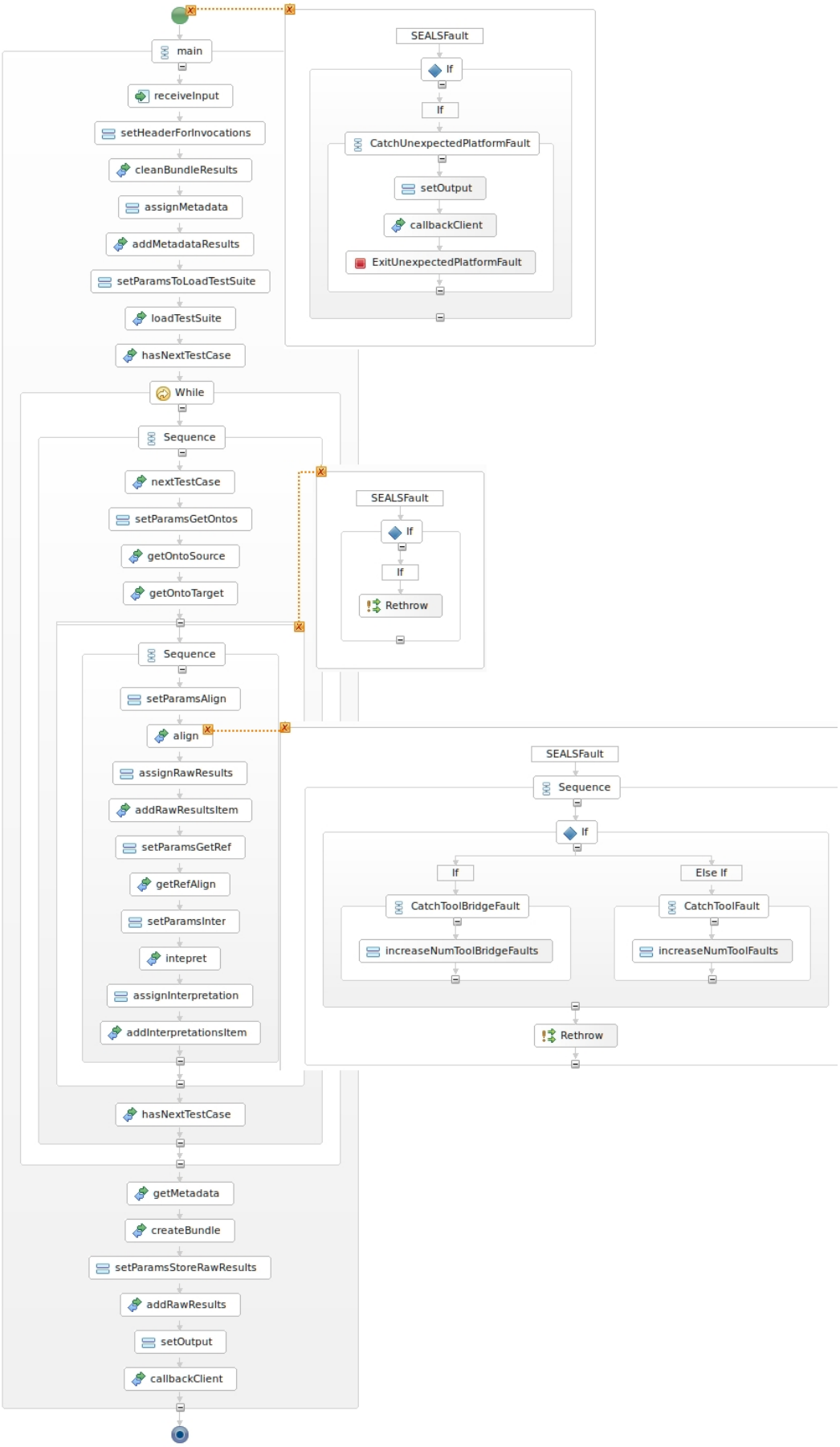


Figure A.1: BPEL workflow for the evaluation of ontology matching tools.

References

[1] Oliver Bodenreider. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, (32):267–270, 2004.

[2] Watson Wei Khong Chua and Jung-Jae Kim. Eff2Match results for OAEI 2010. In *Proceedings of the ISWC 2010 workshop on ontology matching*, Shanghai, China, 2010.

[3] Isabel F. Cruz, Cosmin Stroe, Michele Caci, Federico Caimi, Matteo Palmonari, Flavio, Palandri Antonelli, and Ulas C. Keles. Using agreementmaker to align ontologies for oaei 2010. In *Proceedings of the ISWC 2010 workshop on ontology matching*, Shanghai, China, 2010.

[4] Jérôme David. Aroma results for oaei 2008. In *Proceedings of the ISWC 2008 workshop on ontology matching*, 2008.

[5] Hong-Hai Do, Sergei Melnik, and Erhard Rahm. Comparison of schema matching evaluations. In *Proc. Workshop on Web, Web-Services, and Database Systems*, volume 2593 of *Lecture notes in computer science*, pages 221–237, Erfurt (DE), 2002.

[6] Cássia Trojahn dos Santos, Jérôme Euzenat, Christian Meilicke, and Heiner Stuckenschmidt. D12.1 Evaluation Design and Collection of Test Data for Matching Tools. Technical report, SEALS Project, November 2009.

[7] Jérôme Euzenat. An API for ontology alignment. In *Proceedings of the 3rd International Semantic Web Conference (ISWC)*, pages 698–712, Hiroshima (JP), 2004.

[8] Jérôme Euzenat, Alfio Ferrara, Laura Hollink, Véronique Malaisé, Christian Meilicke, Andriy Nikolov, Juan Pane, Francois Scharffe, Pavel Shvaiko, Vassilis Spiliopoulos, Heiner Stuckenschmidt, Ondrej Sváb-Zamazal, Vojtech Svátek, Cássia Trojahn dos Santos, and George Vouros. Preliminary results of the ontology alignment evaluation initiative 2009. In *OM*, 2009.

[9] Faycal Hamdi, Brigitte Safar, Nobal B. Niraula, and Chantal Reynaud. Taxomap alignment and refinement modules: Results for oaei 2010. In *Proceedings of the ISWC 2010 workshop on ontology matching*, Shanghai, China, 2010.

[10] Wei Hu and Yuzhong Qu. Falcon-ao: A practical ontology matching system. *Journal of Web Semantics*, 6:237–239, 2008.

[11] Prateek Jain, Pascal Hitzler, Amit P. Sheth, Kunal Verma, and Peter Z. Yeh. Ontology alignment for linked open data. In *Proceedings of the International Semantic Web Conference*, 2010.

[12] Ernesto Jimenez-Ruiz, Bernardo Cuenca Grau, Ian Horrocks, and Rafael Berlanga. Towards a umls-based silver standard for matching biomedical ontologies. In *Proceedings of the ISWC 2010 workshop on ontology matching*, Shanghai, China, 2010.

[13] Yoonkyong Lee, Mayssam Sayyadian, AnHai Doan, and Arnon S. Rosenthal. etuner: tuning schema matching software using synthetic scenarios. *The VLDB Journal*, 16(1):97–122, 2007.

[14] Christian Meilicke, Cássia Trojahn, Jérôme Euzenat, and Heiner Stuckenschmidt. Services for the automatic evaluation of matching tools. Technical Report D12.2, SEALS Project, July 2010.

[15] Md. Hanif Seddiqui and Masaki Aono. Anchor-flood: results for OAEI 2009. In *Proceedings of the ISWC 2009 workshop on ontology matching*, Washington DC, USA, 2009.

[16] Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, William Bug, Werner Ceusters, Louis J. Goldberg, Karen Eilbeck, Amelia Ireland, Christopher J Mungall, Neocles Leontis, Philippe Rocca-Serra, Alan Ruttenberg, Susanna-Assunta Sansone, Richard H Scheuermann, Nigam Shah, Patricia L Whetzel, and Suzanna Lewis. The obo foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology*, (25):1251–1255, 2007.

[17] Ondrej Svab, Vojtech Svatek, Petr Berka, Dusan Rak, and Petr Tomasek. Ontofarm: Towards an experimental collection of parallel ontologies. In *Poster Track of ISWC*, Galway, Ireland, 2005.

[18] Cássia Trojahn, Christian Meilicke, Jérôme Euzenat, and Heiner Stuckenschmidt. Automating oaei campaigns (first report). In *Proceedings of the International Workshop on Evaluation of Semantic Technologies (IWEST)*, 2010.

[19] Cássia Trojahn, Christian Meilicke, Jérôme Euzenat, and Ondřej Šváb Zamazal. Results of the first evaluation of matching tools. Technical Report D12.3, SEALS Project, November 2010.

[20] Peng Wang and Baowen Xu. Lily: The results for the ontology alignment contest oaei 2007. In *Proceedings of the ISWC 2007 workshop on ontology matching*, Busan, Korea, 2007.

[21] Zhichun Wang, Xiao Zhang, Lei Hou, Yue Zhao, Juanzi Li, Yu Qi, and Jie Tang. RiMOM results for oaei 2010. In *Proceedings of the ISWC 2010 workshop on ontology matching*, Shanghai, China, 2010.

[22] Peigang Xu, Yadong Wang, Liang Cheng, and Tianyi Zang. Alignment results of sobom for oaei 2010. In *Proceedings of the ISWC 2010 workshop on ontology matching*, Shanghai, China, 2010.